IN THE

Supreme Court of the United States

October Term, 1975
No. 74-1033

C. MARSHALL DANN, COMMISSIONER OF PATENTS & TRADEMARKS,

Petitioner,

vs.

THOMAS R. JOHNSTON,

Respondent.

On Writ of Certiorari to the United States Court of Customs and Patent Appeals.

Brief Amicus Curiae for Software Associates, Inc.

REED C. LAWLOR,
412 West Sixth Street,
Los Angeles, Calif. 90014,
Attorney for Amicus Curiae.

Of Counsel:
LEONARD TACHNER,
412 West Sixth Street,
Los Angeles, Calif. 90014.

Parker & Son, Inc., Law Printers, Los Angeles. Phone 724-6622

SUBJECT INDEX

	Page
I. The Question	1
II. Consent	1
III. Interest of Software Associates, Inc.	2
IV. Summary of Argument	3
V. Comparison of Software Programming and Hardware Programming	4
VI. A Programmed Computer Constitutes Statutory Subject Matter	6
VII. Software Inventions and Hardware Inventions Are Equally Patentable	13
VIII. All the Evidence of Record Supports the Conclusion That Johnston's Programmed Computer Is a New Machine	19
IX. Gottschalk v. Benson Favors Granting of Patents on Inventive Programming of Computers	20
X. Patents for Inventions Implemented by Means of New Programs Take Nothing Away From, but Provide Something New for, Owners of Old Computers	24

XI. Page

The CBEMA Relies on Statements of Foreign Origin That, in Fact, Do Not Support Its Position Against Patentability, but Instead Support Amicus' Position for Patentability 26

XII.

Conclusion 29

TABLE OF AUTHORITIES CITED

Cases Page

Bernhart and Fetter, In re, 417 F.2d 1395 (CCPA, 1969) 6, 7, 23

Brebner, In re, 173 USPQ 169 (CCPA, 1972) .. 18

Chicago Pneumatic Tool Co. v. Hughes Tool Co., 97 F.2d 945 (1938) 18

Christensen, In re, 748 F.2d 35 (1973) 21

Comstock & Gilmer, In re, 481 F.2d 950 (1973) .. 7

Continental Paper Bag Co. v. Eastern Paper Bag Co., 310 U.S. 405 (1908) 17

Foster, Kerns, and Sengbush, In re, 438 F.2d 1011 (1971) 23

Gay, In re, 135 USPQ 311 (CCPA, 1962) 18

Gottschalk v. Benson, 409 U.S. 63 (1972) 3
.....6, 20, 21, 22, 23, 29

Graham v. John Deere, 383 U.S. 1 (1966) 8, 19

Graver Tank & Mfg. Co., Inc., et al. v. The Linde Air Products Company, 339 U.S. 605 17

Great Atlantic & Pacific Tea Co. v. Supermarket Equipment Corp., 340 U.S. 147 (1950) 8, 25

King & Barton, Ex parte, 146 USPQ 590 (Pat. Off. Board of Appeals, 1960) 6

Prater & Wei, In re, 415 F.2d 1393 (CCPA, 1969) 6, 23

Tilghman v. Proctor, 102 U.S. 707 (1880) 17
.....18, 23

United States v. Dubilier Condenser Corp., 289 U.S. 179 (1933) 24

Page

Western Electric Company, Incorporated in the Appeal, the German Federal Court, Case No. 778, 4 Computer Law Service Reports 474 (Callaghan, 1974) 28

Statutes

35 USC 101 .....1, 3, 6, 19

35 USC 102 .....3, 18, 19

35 USC 103 .....3, 4, 6, 14, 18

35 USC 112 .....3, 6, 16, 17, 18, 22, 23

35 USC 115, 25 19

Miscellaneous

Aiken, Howard H., et al., "A Manual of Operation for the Automatic Sequence Controlled Calculator", Harvard Univ. Press, 1946 in The Annals of the Computation Laboratory of Harvard University, Vol. 1, pp. 3-4 21

American Bar Association, 1974 Committee Reports of the Section of Patent, Trademark and Copyright Law, pp. 306-320 21

Aspden, G., remarks at pp. 47-48 of "Software Protection—The Legal Protection of Computer Programs", Report of Proceedings (British Computer Society, 1970) 28

BNA's Patent, Trademark & Copyright Journal, No. 124, "Are Computer Programs Ever Patentable? Did The Patent Office Win?" (BNA, 1973) 21

British Computer Society, "Software Protection—The Legal Protection of Computer Programs", Report of Proceedings (British Computer Society, 1970) 26

Page

Chu, Yoahan, Introduction to Computer Organization (Prentice Hall, 1970) 14, 15, 16

IBM System/360 Operating System, FORTRAN IV (G and H) Progammer's Guide, Form C28-6817-0 (IBM, 1968) 10

Katzan, Harry, Jr., "Computer Organization and the System/370", Computer Science Series (Van Nostrand Reinhold Co., 1971) 14, 15, 16

Knuth, Donald, The Art of Computer Programming, Vol. 1 (Addison, Wesley, 1968) 14, 15

Lawlor, Reed C., "Benson and Beyond", Computer Law Service, Vol. 3-2, Art. 3 (Callaghan & Company, 1975) 21

Parmelee, R. P., et al., "Virtual Storage and Virtual Machine Concepts", IBM Systems J, Vol. II, No. 2 (1972) 14, 15, 16

Robinson, Treatise on the Law of Patents, Vol. 1 (1890) 23

Roehl, Harry N., "Player Piano Treasury" (Vestal Press, 1961) 11

1975 World Intellectual Property Organization Meeting of the Advisory Group of Non-Governmental Experts on the Protection of Computer Programs, Report No. AGCP/NGO/II/4, Annex II, March 12, 1975, and Report No. AGCP/NGO/II/11, Annex II, July 14, 1975 27

IN THE

Supreme Court of the United States

October Term, 1975
No. 74-1033

C. MARSHALL DANN, COMMISSIONER OF PATENTS & TRADEMARKS,

Petitioner,

vs.

THOMAS R. JOHNSTON,

Respondent.

On Writ of Certiorari to the United States Court of Customs and Patent Appeals.

Brief Amicus Curiae for Software Associates, Inc.

I.
The Question.

Whether an invention is unpatentable under 35 USC 101 merely because it has been implemented by programming a modern general purpose digital computer.

II.
Consent.

Software Associates, Inc. has received consent from both parties to file this brief *amicus curiae*. The letters of consent have been filed with the Clerk of the Court.

III.

Interest of Software Associates, Inc.

This Amicus, Software Associates, Inc., is a small corporation located in Altadena, California and engaged in the manufacture of software for a wide variety of customers. Amicus believes that it is typical of thousands of other small companies engaged in software manufacture.

Amicus has no interest in Johnston's patent application and does not consider itself prepared to form a judgment as to whether or not the invention described and claimed there is unobvious over the prior art.

However, Amicus believes that it is qualified to discuss the general proposition as to whether a programmed computer is a different machine from an unprogrammed computer and strongly opposes the Petitioner's position with respect to the first two questions set forth by Petitioner at page 2 of his brief.

Because of the uncertain status of the patent law, at least as it appears to the software industry, Amicus has not sought to protect any of its products under the patent laws. Discouraged, Amicus, like many other similar companies, has sought to rely upon the law of trade secrets and the law of contracts in order to protect whatever proprietary rights that it has.

Amicus is familiar with the benefits that small companies derive from the United States patent laws in other fields. By obtaining patents on their inventions, small companies in other industries are given an opportunity to grow, and sometimes do succeed in growing. Amicus believes that companies engaged in software development deserve to benefit from the patent laws to

the same extent as companies in other industries when they make meritorious inventions.

Amicus is familiar with *Gottschalk v. Benson*, 409 U.S. 63 (1972) and is also familiar with many of the practices of the United States Patent Office. Amicus is confused by *Gottschalk v. Benson, supra*, and is dismayed by the prospective costs and difficulties of obtaining a patent on an invention that is implemented by software programming and especially by the prospects that a patent may not be obtained at all without resort to an expensive appeal proceeding.

Amicus recognizes that implementation of most programs does not involve patentable invention. But Amicus also believes that the implementation of many inventions that are based upon programming, whether it be by software programming or hardware programming, are meritorious and deserving of patent protection.

IV.

Summary of Argument.

Amicus, Software Associates, Inc. urges this Court to recognize that programming of computers either by software methods or by hardware methods, constitutes patentable subject matter under 35 USC 101 and that patents should be granted on the basis of inventions implemented by programming of computers, so long as the requirements of 35 USC 102, 103, and 112 are satisfied.

Software programing of a programmable computer transforms it to a new machine that functions differently from either the program alone or the unprogrammed computer alone. The changes in the machine resulting from programming can be detected in various ways.

One general purpose digital computer can even be software-programmed to function like another general purpose computer. Such programming is called "simulation." This fact demonstrates not only that a software-programmed computer is a different machine from an unprogrammed computer, but also that hardware programming and software programming are equivalents.

Simulation is often employed in the early development of a new general purpose computer before it has been built. The function of a new general purpose digital computer is the same regardless of whether it has been constructed by hard wiring or constructed by simulation. The invention is the same in either case. Under 35 US 103, patentability of an invention does not depend on how it has been made. Hence, a machine is statutory subject matter regardless of whether software programming or hardware programming is used in its construction.

V.
Comparison of Software Programming and Hardware Programming.

The analysis presented here will be aided by some familiarity with the similarities of and the differences between hardware and software programming.

Broadly, in computer technology, the term software program often refers to a series of instructions which are written in numerical form or in alphabetical form. These instructions are then stored in a program, such as a deck of punched cards, and then transferred from the cards to the memory unit of the computer where the program is stored on a temporary, or semi-permanent, basis. On the other hand, when the program is

stored in the memory unit by hardware programming, such as by soldering of wires, the program usually remains stored on a permanent basis.

In modern general purpose digital computers, the memory unit frequently consists of magnetic cores, or rings, through which wires are threaded. To simplify the discussion, we will usually assume that we are dealing with a memory unit in which the various cells consist of magnetic cores which can be magnetized in one direction or in the opposite direction around the core.

A program may also be stored in a paper tape by punching holes in it, or in a magnetic tape by magnetizing successive parts of the tape in different ways to represent holes and non-holes. In any event, the programmed device is fed to a card reader or tape reader. As successive parts of the programmed device are fed into the reader, electric signals are created in accordance with the marks or holes in the cards or tape in order to set the magnetization of the cores in one direction or the other, thus storing the program in the computer. In such a storage process, the changes in magnetization in effect operate to change the interconnections between the various wires in order to permit the flow of current in one direction or to prevent the flow of current in that direction. Various sections of the magnetic memory unit, not necessarily physically contiguous with each other, operate together to bring about certain associated operations. Such sections of the memory unit act as controls to perform different functions.

Software programming and hardware programming are often equivalent and one may be employed in place of the other for many different purposes.

It is to be noted that Johnston's application describes his invention as being embodied in a computer that has been transformed in accordance with his program. But Johnston's claims do not specify a program. They specify only the machine formed by programming in terms of means plus functions, as authorized by 35 USC 112.

VI.

A Programmed Computer Constitutes Statutory Subject Matter.

When a program is stored in a general purpose digital computer, the states of various memory elements in the computer are transformed. If the program is new, the pattern of states is new, thus creating a new machine that performs new functions.

New machines constitute statutory subject matter (35 USC 101) regardless of how they are made (35 USC 103).

While the Petitioner, the Commissioner of Patents and Trademarks, recognizes that a new machine constitutes statutory matter (Pet. Br.* p. 11), he urges that programming a general purpose computer does not create a new machine (Pet. Br. pp. 29-38). Case after case has held that programming of a general purpose digital computer creates a new machine. *Ex parte King and Barton*, 146 USPQ 590, 591 (Patent Office Board of Appeals, 1960), *In re Prater and Wei*, 415 F.2d 1393, 1405 (CCPA, 1969), *In re Bernhart and Fetter*,

*The following abbreviations are employed in this brief: "Pet. Br." refers to brief of the Petitioner; "A" refers to the Appendix; "CBEMA" refers to the Computer and Business Equipment Manufacturers Association; "CBEMA Br." refers to the brief of the CBEMA: "Tr. Oral Hearing" refers to the oral hearing held on October 16, 1972 in *Gottschalk v. Benson, supra*.

417 F.2d 1395, 1400 (CCPA, 1969), *In re Comstock and Gilmer*, 481 F.2d 905, 909 (CCPA, 1973). In spite of this long-standing recognition that a programmed general purpose digital computer constitutes a different machine from the unprogrammed computer, the Petitioner now contends the contrary (Pet. Br. p. 12), but supplies no evidence to support his contention.

Because of Petitioner's contention, it is necessary to discuss the issue as to whether a programmed computer constitutes a new machine, without an opportunity to take evidence on the question. The Petitioner has given no proof that a programmed computer is not a new machine. Amicus will attempt to persuade the Court that it is, by reference to evidence that is available to the Court.

When the memory unit of the computer includes magnetic cores interlinked by wires, feeding a new program into the computer, in effect, causes the wires to be interconnected in a new way. After a new program has been stored in such a computer by setting the magnetization of the cores, the various parts of the computer are then in a new set of conditions. As a consequence, they coact or function in a new way, so as to produce new results. While some parts of the computer continue to operate as they did before, by adding, subtracting, or performing other preset functions, the programming of the computer changes other parts to create new controls, so that new sequences of operations and other actions occur. New signals are created because of the new interactions. The Petitioner overlooks those new interactions (See for example, Pet. Br. pp. 16-17). And it is these new interactions that makes the programmed computer a new combination of parts with new controls, as distinguished from an ag-

gregation that does nothing new. *Graham v. John Deere Co.*, 383 U.S. 1 (1966). *Great Atlantic & Pacific Tea Co. v. Supermarket Equipment Corp.*, 340 U.S. 147 (1950).

Though a programmed general purpose digital computer looks the same as an unprogrammed general purpose digital computer, after being programmed it is not the same, and it no longer functions merely in the same old way. Its new functions constitute evidence that it is a new machine. There are numerous ways that a person can satisfy himself that a programmed computer has been transformed into a new machine by the program fed into it.

One way is to reflect on what the program can do without the computer and what the computer can do without the program. We will reflect below with reference to the action of Johnston's program in relationship to the IBM 1400 to which Johnston's program has been applied in his description.

1. First, consider a program that is represented by a deck of punched cards. Without a computer it is useless for its purpose. The program deck will not calculate.
2. Now consider the 1400. Unprogrammed, it will produce no useful result at all, no matter how many times you push its START button. Until it has been changed by implementation of Johnston's program in it, it will not produce a result for which the program was designed.
3. Now let's program the computer by feeding Johnston's program deck into it. During the feeding operation, the computer takes the cards from the deck, one at a time, and transforms

the patterns of holes on the cards into a new pattern of states of magnetic cores (or other memory elements) of the memory unit of the computer, to store a magnetic or electrical counter part of the program. The cards are not consumed, but remain intact for later reuse.
It is only after the 1400 has been transformed by storing Johnston's program in its memory unit that it is capable of being operated to achieve Johnston's results.

4. Now let's feed records of customer data into the programmed computer as taught by Johnston, and push the START button. This time Johnston's new results are produced. The records are processed automatically in accordance with the procedures carried out in the action of the new controls created by Johnston's program. After the computer has been programmed as taught by Johnston, it remains altered indefinitely (so long as Johnston's stored program is not removed), so that it can continue to achieve Johnston's results each time new data is fed to it.

The reason that the 1400 produces Johnston's new results when it has been programmed by Johnston, is that the machine has been changed to function in a new way. Thus it is clear that the programmed machine is different from the unprogrammed machine, and is indeed a new machine.

Still another way to test a programmed general purpose computer to ascertain that it is indeed a new machine, is to make tests of standard kinds to determine what the states are of the various parts of the computer after it has been programmed.

The states of the memory elements of the programmed computer can actually be pictured by a process known as a "DUMP".[1] Such a process involves electrically triggering the memory elements to create a stream of signals that represents the states of the memory elements and using that stream of signals to print numbers representative of those states on a strip of paper. These numbers provide the operator of the machine with a "picture" or "snapshot" of the *pattern* of states. A succession of "snapshot" DUMPS can even be made while the computer is operating in a "slow motion" mode.

A skilled programmer can read a succession of such snapshots, to ascertain the new physical interrelations that have been created by the program between various parts of the computer and their reactions to the input signals and to each other to produce output signals.

From the foregoing explanation, it is clear that (1) the action of a new program on a computer is synergistic and that a new machine is created by that synergism, (2) the interdependence between the modified parts of the resultant machine is also synergistic, and the coaction is different from that which existed before, and (3) the results may also be synergistic.

Still another way of recognizing that programming an unprogrammed computer creates a different machine, is set forth in the next section.

The Petitioner and the CBEMA have attempted to belittle the programming of a general purpose digital

[1]*IBM System/360 Operating System, FORTRAN IV (G and H) Programmer's Guide*, Form C28-6817-0, pp. 80-81 (IBM, 1968).

computer by comparing it unfavorably to the playing of a piano roll in a player piano and similar reproducing devices (Pet. Br. pp. 16, 35, CBEMA Br, p. 6). The comparison is incomplete and neglects the differences. We will consider the player piano comparison.

To emphasize the difference between a reproducing player piano that plays a perforated piano roll and a programmable general purpose digital computer, we will consider a general purpose digital computer that is programmed by means of two punched tapes, one that bears holes that represent the program and the other that bears holes that represents data.

A player piano, indeed, produces sounds at its output corresponding to the holes on punched roll, line for line. But that is not what is done by a programmed general purpose digital computer.

The computer is altered semi-permanently by feeding in signals corresponding to the holes in the program tape. Then signals corresponding to data are fed in from another punched tape or otherwise. And then the programmed computer manipulates the data signals in a manner controlled by the programmed computer to produce an output which is far different from any line-by-line reproduction of the input data.

A player piano is designed to reproduce as precisely as possible the music played by a musician.[2] The less deviation there is from the original, in the reproduction by the player piano, the better the player piano is considered. The very nature of the action of the player piano remains the same from roll to roll.

A player piano is merely a reproducing machine with one fixed rule of action. That rule of action is

[2]Roehl, Harry N., "Player Piano Treasury", pp. 46-81 (Vestal Press, 1961).

not changed by reading the punched roll. A programmed computer is a signal processing machine in which the rule of action depends upon how the computer has been programmed, so that data later supplied from a punched tape or the like is processed differently according to the nature of the controls for which the computer has been programmed.

With Johnston's programmed computer, unlike with a player piano, the output is not a reproduction of the input data, but is a new composition generated by the computer to meet the requirements of the customer who supplied the data. If different customers supply the same data, the reports generated will be different when the customer requirements are different.

At page 5 of its brief, the CBEMA seeks to distinguish between application programs and system programs. The CBEMA has not explained the difference satisfactorily. The main difference seems to be that "system programs" are normally developed by computer manufacturers, whereas application programs are generally created by others. The CBEMA seeks one patentability standard for application programs and another for system programs.

All programs that alter a memory unit of a general purpose computer to control it to achieve new results, affect the interaction of the parts. All such programs alter the computer so that it achieves new results. So there is no real dichotomy between system programs and the application programs.

Petitioner says that the patentability of inventions based on system programs is not at issue in this case (Pet. Br. p. 16). But the questions 1 and 2 that he presents to be decided at page 2 of his brief are not

limited to only one kind of program. The Patent Act draws no distinction between such programs. None is deducible from it, and there is no clear semantic difference between all system programs and all application programs. It follows that a computer that has been transformed by a program, constitutes a new machine regardless of whether it involves a "system program" or an "application program". And Johnston's invention relates to a system (Pet. Br. p. 6, and lines 1, 2, 5 and 27 of his claim 20 (A., p. 107)).

That a programmed computer is a different machine from an unprogrammed computer is no legal fiction. And two differently programmed computers are different machines. And as we will see, one general purpose digital computer can be software programmed to function like another even though their hardware is different.

VII.

Software Inventions and Hardware Inventions Are Equally Patentable.

In this section, Amicus desires to call the Court's attention to some well-established principles of computer technology that go under the names of "simulation" and "emulation". These words are terms of art that have special meanings in the field of computer technology, somewhat different from their dictionary meanings. The essence of both "simulation" and "emulation" is that one general purpose digital computer may be so altered by programming that it functions like a different general purpose digital computer. Both "simulation" and "emulation" involve the use of software programming in which programs are stored in memory units by transformation of the states of the various cells in those

memory units. On the other hand, emulation also makes use of some additional physical components.

Under 35 USC 103, patentability is not to be negatived by the manner in which an invention is made. It therefore follows that if a new machine is created by software programming it is just as patentable as if it were made by hardware programming.

Since a new general purpose digital computer that has been hardware programmed constitutes statutory subject matter, Amicus urges that if a new general purpose digital computer has been made by software programming, it is also statutory subject matter. And that therefore, all programmed general purpose digital computers constitute statutory subject matter.

Let us explain more fully.

One old general purpose computer, known as a host computer, can be software programmed to function like a different old general purpose computer, called a target, or simulated, computer. When a target computer has been simulated on a host computer, the simulation is known as a "virtual" computer. The virtual computer can then execute user's programs and data in the same way as the target computer and can then produce the same results as the target computer.[3]

[3]Knuth, Donald, *The Art of Computer Programming*, Vol. 1, page 199 (Addison Wesley, 1968). Chu, Yoahan, *Introduction to Computer Organization*, page 170 (Prentice Hall, 1970). Katzan, Harry, Jr., "Computer Organization and the System/370", *Computer Science Series*, pages 264-265 (Van Nostrand Reinhold Co., 1971). Parmelee, R. P., *et al.*, "Virtual Storage and Virtual Machine Concepts", pp. 99-130, *IBM Systems J*, Vol. II, No. 2 ,1972).

It is not only possible to simulate an existing target computer on a host computer, it is even possible to simulate on a host computer, a newly invented computer that has not yet been actually built by being wired and bolted together, and made ready for use.[4] Professor Knuth has pointed out that it is usually advantageous for a computer manufacturer to simulate a new machine before it has been built, so that software for the new machine may be developed (on the virtual machine) as soon as possible.[5]

It is also possible to simulate several different target computers on the same host general purpose computer at the same time. In this case, users who have programs that were designed to run only on different target machines can run their programs simultaneously on the two virtual machines that they both "see" concurrently embodied in the same host machine.[6]

Since a host general purpose digital computer can be programmed to function like a target general purpose digital computer, it follows that the resultant virtual computer is equivalent to the unprogrammed simulated target computer. Though the virtual computer may be slower in operation than the target computer, the loss of speed is often far outweighed by the saving in the cost of building or buying the target computer.

The equivalence between hardware programming and software programming also becomes readily apparent by

[4]Chu, *supra;* Knuth, *supra*, Katzan, *supra.*

[5]Knuth, p. 199.

[6]Parmelee, *et al., supra.*

considering how an old, host general purpose computer can be restructured to function like a new, target general purpose computer. In order to achieve a restructuring of the old computer to function like the new, three general methods are available, namely:

1. Rewiring the old general purpose digital computer.
2. Programming the old general purpose digital computer to simulate the new general purpose digital computer.
3. Emulating the new general purpose digital computer.

Both simulation and emulation involve software programming; but as previously mentioned, emulation also involves the addition of auxiliary electro-mechanical components.[7]

Of course, to design and develop a new general purpose computer, it is not necessary to construct the new computer in its entirety by programming an old computer. It may be sufficient in many cases merely to simulate some of the important subcombinations of the elements that are to form the new computer.[8]

In any event, when one computer is restructured by rewiring, simulation, or emulation, the modified host computer then functions like the target computer (except possibly for speed of operation). Since the restructured computer functions like the target computer in each case, they are equivalent. It follows therefore, under 35 USC 112, that the target computer is covered by the claims of a properly written patent on the target computer regardless of how the restructuring is accomplished. Infringement would exist regardless of whether

[7]Katzan, *supra;* Parmelee, *et al., supra.*

[8]*Chu, supra.*

the restructuring is accomplished entirely by rewiring or entirely by software programming or partly by each. If this were not so, patents based upon the disclosure of a hardware implementation "would be of little worth". *Continental Paper Bag Co. v. Eastern Paper Bag Co.,* 210 U.S. 405, 418 (1908). *Graver Tank & Mfg. Co., Inc., et al. v. The Linde Air Products Company,* 339 U.S. 605 (1950).

By the same token, an invention that has been made by means of software programming should be patentable on the basis of a disclosure of the software implementation and the patent thereon should cover equivalent hardware.

The fact that one general purpose computer can be constructed by simulation on another computer, can only mean that inventions that have been implemented by software programming are patentable just as are inventions that have been implemented by hardware programming.

Furthermore, if, as the Petitioner sometimes urges in many places (see for example, Pet. Br. pp. 29-38), software programming cannot form the basis of a patentable invention as a machine, then neither can hardware, if that hardware is capable of being simulated by equivalent software. And no one urges that patents may not be based on hardware implementations of programs. It follows therefore that patents may be based on software implementations of programs.

Under 35 USC 112, it is only necessary for an Applicant for patent to describe the best mode contemplated by him at the time for carrying out his invention. It is not necessary to describe any other mode. *Continental Paper Bag v. Eastern Paper Bag Co.,* 210 U.S. 405 at 418-420 (1908). *Tilghman v. Proctor,* 102

U.S. 707 at 726 (1880). *Chicago Pneumatic Tool Co. v. Hughes Tool Co.*, 97 F.2d 945 (CA 10, 1938) cert. den. *In re Gay*, 135 USPQ 311 (CCPA, 1962). *In re Brebner*, 173 USPQ 169 (CCPA, 1972). Accordingly, if an invention is capable of being implemented in both software form and in hardware form, a disclosure of either is sufficient to form a basis for claims that cover both.

Very frequently, software programming of a general purpose digital computer to implement an invention represents the best mode of practicing the invention. This occurs, for example, where it would be too expensive or time-consuming or both, to build a completely new computer by wiring and bolting. It also may represent the best mode where it is desirable to practice the invention at many different locations, but for only short times at each. By software programming, the investment of time and money for achieving the desired results may be greatly reduced and the invention made available more readily and quickly to members of the public. An applicant should be entitled to "means plus functions" claims under 35 USC 112 even if he discloses implementation of his program by software methods. The law requires him to disclose his software methods if that is his best mode. And he should not be punished for complying with the law.

Therefore, inventions that have been implemented by software-programming of computers should be recognized as constituting statutory subject matter and should be held to be patentable so long as the invention satisfies the requirements of 35 USC 102 and 103 and the description and claims satisfy the requirements set forth in 35 USC 112, and the claims do not suffer from the defects of the Benson claims.

VIII.

All the Evidence of Record Supports the Conclusion That Johnston's Programmed Computer Is a New Machine.

The statements of the Petitioner that programming of a general purpose digital computer does not make the computer a new machine, is not supported by evidence. It is merely a categorical, unsupported conclusion.

Johnston's patent application constitutes a document filed under oath or declaration (35 USC 115, 25). Johnston's application, therefore, constitutes evidence. Johnston describes the functions of the computer having his program stored in it. There is no denying that his computer is a machine or that it functions as described. The only question is whether it is a new machine.

Johnston's claims have been rejected as being obvious over the prior art. This rejection constitutes an admission by the Petitioner that Johnston's machine is different from any machine in the prior art. *Graham v. John Deere*, 383 U.S. 1, 17 (1966). Since the rejection is under 35 USC 103 but not under 35 USC 102, Johnston's machine must be a different, and hence new, machine.

There is also no denying that Johnston's programmed computer is useful.

35 USC 101 only requires that a machine be new and useful in order to constitute statutory subject matter.

By the Petitioner's own admission, therefore, it follows that Johnston's claims define statutory subject matter under 35 USC 101.

IX.

Gottschalk v. Benson Favors Granting of Patents on Inventive Programming of Computers.

The majority of the Court below held that the Johnston invention constituted statutory subject matter and that the invention was properly claimed.

In the sole dissenting opinion concerning the issue of statutory subject matter in *In re Johnston* (A., pp. 17ff), the view was expressed by Judge Rich that *Gottschalk v. Benson, supra,* should be clarified. Amicus agrees with the majority opinion that the subject matter of Johnston's claims constitutes statutory subject matter and that the invention was properly claimed. Amicus also agrees with the dissenting opinion of Judge Rich that *Gottschalk v. Benson, supra,* should be clarified.

At some points, Petitioner concedes that programming of a computer may constitute patentable subject matter (Pet. Br. p. 29). But at other points (*e.g.*, Pet. Br. pp. 28, 35) Petitioner is inconsistent and, Amicus believes, in error.

More particularly, at some points of his brief, the Petitioner seems to proceed on the assumptions (1) that *Gottschalk v. Benson, supra,* held that no process claim to an invention that is implemented by programming of a general purpose digital computer is patentable (Pet. Br. pp. 7, 8, 9, 12, 27, and 28); (2) that apparatus claims are equivalent to process claims in every respect (Pet. Br. pp. 7, 10, 31); and (3) that therefore all apparatus claims to inventions that are implemented by software programming of a general purpose digital computer are also non-statutory (Pet. Br. pp. 8, 12, and 29). With these three propositions, Amicus disagrees.

The holding of *Gottschalk v. Benson, supra,* was a narrow one. *In re Cristensen,* 478 F.2d 1392 (CCPA, 1973).[9] The *Benson* holding was made with respect to process claims that had failed to satisfy any of four important criteria. Thus, in *Benson,*

1. The claims were not limited to any particular art or technology, AND
2. The claims were not limited to any particular apparatus or machinery, AND
3. The claims were not limited to any particular end use, AND
4. Furthermore, they were not limited to "transformation and reduction of an article 'to a different state or thing' ".

As a result, Benson's claims were broad enough to cover a mental process for carrying out a mathematical algorithm. That's what the case was all about (Tr. Oral Hearing, p. 30).

It has been known since at least as early as 1944 that the process by which even the most simple algorithms are carried out in a general purpose digital computer differs from mental processes for carrying out those same operations.[10] The Benson, *et al.* claims

---

[9]Lawlor, Reed C., "Benson and Beyond", Computer Law Service, Vol. 3-2 Art. 3 (Callaghan & Company, 1975). (A copy is on file with the Clerk.) See also, "Are Computer Programs Ever Patentable? Did The Patent Office Win?", *BNA's Patent, Trademark & Copyright Journal,* No. 124, pp. C-1 to C-6, 4-19-73 (BNA, 1973). A comprehensive bibliography on computer program protection will be found in the 1974 Committee Reports of the Section of Patent, Trademark and Copyright Law of the American Bar Association, pp. 306-320.

[10]Aiken, Howard H., *et al.*, "A Manual of Operation for the Automatic Sequence Controlled Calculator", Harvard Univ. Press, 1946, in *The Annals of the Computation Laboratory of Harvard University,* Vol. 1, pp. 3-4.

were not even limited to carrying out their process in a digital computer.

While Benson's claim 8 mentioned a "reentrant register", Benson, *et al.* had urged during the prosecution of their application in the Patent Office that a claim specifying a "reentrant register" could be carried out with only pencil and paper. Claim 13 specified no element bearing any resemblance to apparatus. Thus the Benson, *et al.*, claims were not limited to any particular machine, as held by the Court. Not being limited otherwise to satisfy any other criteria mentioned above, it is apparent that Benson, *et al.* were seeking a patent that would cover a mental process that could be carried out with only pencil and paper. In *Benson* it was only such broad claims that were held to be unpatentable.

In *Benson*, the Court clearly indicated that patents for programming of computers, on the basis of either process inventions or machine inventions, were not foreclosed, if the inventions were properly claimed. Thus, the Court implied that inventions that are implemented by programming of computers may not only be claimed as apparatus but even as processes if properly limited.

In contrast, with the claims of Benson, *et al.*, Johnston's claims specify "means plus function", making it clear that his claims are for apparatus and hence are not for a process to be carried out without apparatus. Thus it is clear that Johnson does not consider his invention to lie in a mental process or in an algorithm, but in a machine. Consequently, under 35 USC 112, Johnston's claims cover only the machine that he has

disclosed and equivalents thereof. Since the human mind is not a machine or any part thereof, Johnston's apparatus claims do not cover the human mind or its operations. *In re Prater and Wei*, 415 F.2d 1393, 1406 (CCPA, 1969). *In re Bernhart and Fetter*, 417 F.2d 1395, 1401 (CCPA, 1969), *In re Foster, Kerns, and Sengbush*, 438 F.2d 1011, 1015 (CCPA, 1971). Johnston's claims cover only a machine; and though a machine may perform a process, it is really only a "thing" that is visible to the eye (*Tilghman v. Proctor*, 102 U.S. 707, 728 (1880)).

Equating apparatus claims to process claims, as attempted by the Petitioner (Pet. Br. pp. 7, 10, and 31) is improper. While it is true that the skilled draftsman can write claims either in method form or in apparatus form in certain types of cases, this does not make the two types of claims equal. It has long been recognized that process claims and apparatus claims cover different inventions. One reason is that the process may be invented by one person and the apparatus by another (*Robinson, Treatise on the Law of Patents*, Vol. 1, p. 250). Even when the method and apparatus inventions are made by the same person, the rule is the same. Hence, process claims and apparatus claims differ not only in format, but in substance.

Therefore holding that Johnston's apparatus claims cover statutory subject matter is consistent with *Gottschalk v. Benson, supra*, especially since they specify particular apparatus in the manner set forth in 35 USC 112 and are free of other defects present in the claims of Benson, *et al.*

X.

Patents for Inventions Implemented by Means of New Programs Take Nothing Away From, but Provide Something New for, Owners of Old Computers.

The Petitioner seems to argue, at page 35 of his brief, that the grant of a patent to an invention that involves changing an old computer into a new one by means of a program, would take something away from the public which already possesses the old computer. The fact is that the public would still have a right to use the old computer in the same old way that it always had and even new unpatented ways. The public would only be excluded from practicing the invention in the new programmed computer. This is true regardless of whether hardware programming or software programming is used to achieve the change. The fact that software programming results in only temporary change makes the principle more applicable than if permanent or semi-permanent hardware changes were made, since with software programming, the changed computer can be restored to its original state more easily, swiftly, and economically than if hardware changes were made.

In *United States v. Dubilier Condenser Corp.*, 289 U.S. 178 (1933), the Court stated, at page 186:

> "An inventor deprives the public of nothing which it enjoyed before his discovery, but gives something of value to the community by adding to the sum of useful knowledge."

It is sometimes argued that the patentee would take something away from the public if a patent is issued on the basis of an invention which would involve programming a computer in its practice, because the owner of the computer already has a right to use that program-

based invention in his computer (Pet. Br. p. 35 and CBEMA Br. p. 21). It is argued that computers are intended to use programs; therefore, the public should be entitled to use any new program in spite of the fact that the program was not invented by the manufacturer or owner of the computer. A more unjustified assertion of rights of machine builders and owners could not be made.

To appreciate how unjustified the Petitioner's position is, let us consider the reverse fact situation where the program has been developed and fully tested in one computer prior to the invention of another computer.

Suppose that you have discovered a new program and have marketed it for many years. Now suppose that the ABC company later invents a new computer that can use your program more effectively.

Wouldn't it be unfair to argue that you have a right to make and use the new computer without the permission of the ABC company since the ABC company should have no right to restrict your right to use *your* program on *its* new machine? Wouldn't it be unfair to argue that a patent on the new computer should not issue because it allegedly would diminish your right to use your program?

A patent granted to the ABC company on its new computer would take nothing away from you that you already had.

And so it is, granting a patent to the inventor of an implementation of a new program is perfectly proper under the patent law, since he takes nothing away from but adds to the sum of the public's useful knowledge. *Great Atlantic & Pacific Tea Co. v. Supermarket Equipment Corp.*, 340 U.S. 147 (1950).

XI.

The CBEMA Relies on Statements of Foreign Origin That, in Fact, Do Not Support Its Position Against Patentability, but Instead Support Amicus' Position for Patentability.

Time and space do not permit review and discussion of all the incomplete and unsupported statements in the CBEMA brief allegedly based on foreign views regarding inventions of the kind under discussion here. Such statements are really immaterial and irrelevant to the present case.

Nevertheless, since the CBEMA has raised such questions, Amicus desires to point to a few examples that demonstrate that patentability of software programming inventions is favored under foreign views.

At page 25 of its brief, the CBEMA refers to another possible form of protection that has been suggested for preserving some proprietary rights in computer programs (See also Pet. Br. p. 36). These statements might lead one to believe that the Advisory Group of the World Intellectual Property Association has recommended that a new form of protection be developed as a substitute for patent protection. The opposite is the case. Such a proposal (known as the IBM registration proposal),[11] while it may be useful in a limited way, cannot substitute for patent protection of meritorious inventions.

---

[11]"Software Protection, the Legal Protection of Computer Programs", two day conference at the Bedford Hotel, Brighton, Thursday and Friday, 13th and 14th November, 1969. Report of Proceedings, p. 48 (British Computer Society, May 22, 1970). (A copy of those proceedings has been filed with the Clerk of the Court.)

The Advisory Group of the World Intellectual Property Association cited by the CBEMA, has recognized the importance of preserving the rights to obtain patents on program-related inventions, as follows:

> "Patent protection should not be withheld from inventions for the reason that they involve computer programs when such inventions fulfill the traditional requirements of patentability".[12]

This view was reiterated in the final report of the Advisory Committee dated July 14, 1975, where it said,

> "In the discussion on existing forms of legal protection, it was stressed once again that inventions fulfilling the requirements of patentability should not be excluded from protection merely because they related to a computer program or involved a software element. It was noted that patent protection could not apply to computer programs which do not meet the standard of inventiveness. . . ."[13]

and added,

> "as regards legal protection of computer programs, a special type of protection should be established, without prejudice to the continuation of any existing forms of protection, in particular copyright, patent or unfair competition laws: . . .".[14]

---

[12]1975 WIPO (World Intellectual Property Organization) Meetings, Report AGCP/NGO/II/4, dated March 14, 1975, Annex II, page 8, March 12, 1975.

[13]1975 WIPO Meeting, Report AGCP/NGO/II/11, dated July 14, 1975, Annex II, page 3, July 14, 1975.

[14]1975 WIPO Meeting, Report AGCP/NGO/II/11, *supra*, page 5.

The brief of the CBEMA also cites instances of actions in foreign countries that purportedly would be unfavorable to Johnston's position if they were U.S. cases. In Germany, with respect to which CBEMA has cited two vaguely identified and probably inapposite cases, in a third case, not cited by the CBEMA, the Senate of the German Federal Patent Court, which is the highest patent court in Germany, held that claims in the German Benson case that were narrower than those in the U.S. Benson case, were patentable under German law, and ordered the patent published.[15]

The CBEMA brief also refers to the effect that U.S. patents might have on trade.

The recognition that inventions implemented by programming of computers constitute patentable matter, would actually increase competition and public disclosure of important programs. It would compel hardware manufacturers to disclose their important programs in patents instead of keeping them secret. At the same time, recognition that such inventions are patentable, would facilitate growth of small software manufacturers, all of which appears to be opposed by many hardware manufacturers.[16]

---

[15]In the appeal, Western Electric Company Incorporated, Case No. 778, the German Federal Patent Court, May 28, 1973. An English translation of the opinion appears at 4 *Computer Law Service Reports*, p. 474 (Callaghan, 1974).

[16]These views were explained by Dr. G. Aspden, Director of European Patent Operations IBM, UK Laboratories, Ltd., in a panel discussion of a conference held by the British Computer Society in 1969. See "Software Protection, the Legal Protection of Computer Programs", *supra* at pp. 47-48.

XII.

Conclusion.

For the above reasons, Amicus urges this Court to clarify *Gottschalk v. Benson*, and particularly to hold that an invention that is implemented by programming a general purpose digital computer constitutes statutory subject matter under 35 USC 101.

Respectfully submitted,

REED C. LAWLOR,
Attorney for Amicus Curiae.

Of Counsel:
LEONARD TACHNER.